Structured Output Prediction: Discriminative Training

CS4780/5780 – Machine Learning
Fall 2019

Nika Haghtalab & Thorsten Joachims
Cornell University

Reading: Murphy 19.7

Structured Output Prediction

• Supervised Learning from Examples
  – Find function from input space X to output space Y
    \[ h: X \rightarrow Y \]
  such that the prediction error is low.

• Typical
  – Output space is just a single number
    • Classification: -1,+1
  – Regression: some real number

• General
  – Predict outputs that are complex objects

Training HMMs with Structural SVM

• HMM
  \[ P(x,y) = P(y_1)p(x_1|y_1) \prod_{i=2}^{T} p(x_i|y_i)p(y_i|y_{i-1}) \]
  \[ \log P(x,y) = \log P(y_1) + \log p(x_1|y_1) + \sum_{i=2}^{T} \log p(x_i|y_i) + \log P(y_i|y_{i-1}) \]

• Define \( \phi(x,y) \) so that model is isomorphic to HMM
  – One feature for each possible start state
  – One feature for each possible transition
  – One feature for each possible output in each possible state
  – Feature values are counts

Joint Feature Map for Sequences

• Linear Chain HMM
  – Each transition and emission has a weight
  – Score of a sequence is the sum of its weights
  – Find highest scoring sequence \( h(x) = \arg \max_{y \in Y} [w \cdot \phi(x,y)] \)

Joint Feature Map for Trees

• Weighted Context Free Grammar
  – Each rule \( \tau_i \) (e.g. \( S \rightarrow NP VP \)) has a weight
  – Score of a tree is the sum of its weights
  – Find highest scoring tree \( h(x) = \arg \max_{y \in Y} [w \cdot \phi(x,y)] \)

Idea for Discriminative Training of HMM

Idea:

\[ h_{Bayes}(x) = \arg \max_y \{ P(Y = y | X = x) \} \]

\[ = \arg \max_y \{ P(X = x | Y = y) P(Y = y) \} \]

– Model \( P(Y = y | X = x) \) with \( \Vec{w} \cdot \phi(x,y) \) so that

\( \{ \arg \max_y \{ P(Y = y | X = x) \} \} = \{ \arg \max_y \{ \Vec{w} \cdot \phi(x,y) \} \} \)

Hypothesis Space:

\[ h(x) = \arg \max_y \{ \Vec{w} \cdot \phi(x,y) \} \]

with \( \Vec{w} \in \Re^{|Y|} \)

Intuition:

– Tune \( \Vec{w} \) so that correct \( y \) has the highest value of \( \Vec{w} \cdot \phi(x,y) \)

– \( \phi(x,y) \) is a feature vector that describes the match between \( x \) and \( y \)

Joint Feature Map for Sequences

<table>
<thead>
<tr>
<th>The dog chased the cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>N \rightarrow V</td>
</tr>
<tr>
<td>V \rightarrow Det</td>
</tr>
<tr>
<td>Det \rightarrow N</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>\phi(x,y) = [w \cdot \phi(x,y)]</td>
</tr>
</tbody>
</table>

Joint Feature Map for Trees

<table>
<thead>
<tr>
<th>The dog chased the cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>NP \rightarrow N</td>
</tr>
<tr>
<td>N \rightarrow dog</td>
</tr>
<tr>
<td>Det \rightarrow the</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>V \rightarrow chased</td>
</tr>
<tr>
<td>N \rightarrow cat</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>\phi(x,y) = [w \cdot \phi(x,y)]</td>
</tr>
</tbody>
</table>
Structural Support Vector Machine

- Joint features $\phi(x, y)$ describe match between $x$ and $y$
- Learn weights $w$ so that $w \cdot \phi(x, y)$ is max for correct $y$

**Hard-margin optimization problem:**

$$\min_{w} \frac{1}{2} w^T w \\
s.t. \quad \forall y \in Y \setminus Y_1 : w^T \Phi(x_1, y_1) \geq w^T \Phi(x_1, y) + 1$$

**Soft-margin optimization problem:**

$$\min_{w, \xi} \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i \\
s.t. \quad \forall y \in Y \setminus Y_1 : w^T \Phi(x_1, y_1) \geq w^T \Phi(x_1, y) + \Delta(y_1, y) - \xi_i$$

**Lemma:** The training loss is upper bounded by

$$Err(y(h)) = \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, h(x_i)) \leq \frac{1}{n} \sum_{i=1}^{n} \xi_i$$

Soft-Margin Structural SVM

- Loss function $\Delta(y_i, y)$ measures match between target and prediction.

Cutting-Plane Algorithm for Structural SVM

- Input: $(x_1, y_1), \ldots, (x_n, y_n), C, \varepsilon$
- $S \leftarrow \emptyset$, $\varepsilon' \leftarrow 0$, $\xi \leftarrow 0$
- **REPEAT**
  - FOR $i = 1, \ldots, n$
    - compute $\hat{y} = \arg\max_{y \in Y} (\Delta(y, \hat{y}) + w^T \Phi(x, \hat{y}))$
    - IF $(\Delta(y, \hat{y}) - w^T \Phi(x, \hat{y})) > \xi$
      - $S \leftarrow S \cup \{w^T \Phi(x, \hat{y}) - \Delta(y, \hat{y}) - \xi\}$
    - $\xi + = \xi + \varepsilon$
  - $S \leftarrow S \cup \{w^T \Phi(x_1, y_1) - \Delta(y_1, y_1) - \xi\}$
  - $\xi \leftarrow \xi + \varepsilon$
  - **ENDIF**
- **ENDFOR**
- **UNTIL** $S$ has not changed during iteration

Generic Structural SVM

- Application Specific Design of Model
  - Loss function $\Delta(y_1, y)$
  - Representation $\Phi(x, y)$
  - Markov Random Fields [Lafferty et al. 01, Taskar et al. 04]
- Prediction:

$$\hat{y} = \arg\max_{y \in Y} (\hat{w}^T \Phi(x, y))$$

- Training:

$$\min_{\hat{w}, \xi} \frac{1}{2} \hat{w}^T \hat{w} + c \sum_{i=1}^{n} \xi_i$$

$$s.t. \quad \forall y \in Y \setminus Y_1 : \hat{w}^T \Phi(x_1, y_1) \geq \hat{w}^T \Phi(x_1, y) + \Delta(y_1, y) - \xi_i$$

Applications: Parsing, Sequence Alignment, Clustering, etc.
Polynomial Sparsity Bound

- Theorem: The sparse-approximation algorithm finds a solution to the soft-margin optimization problem after adding at most
  \[ \frac{4CA^2R^2}{\varepsilon^2 S} \]
  constraints to the working set, so that the Kuhn-Tucker conditions are fulfilled up to a precision \( \varepsilon \). The loss has to be bounded \( 0 \leq \Delta (y, y) \leq A \), and \( ||\phi(x, y)|| \leq R \).

Applying StructSVM to New Problem

- Basic algorithm stays the same (e.g. SVM-struct)
- Application specific
  - Loss function \( \Delta (y, y) \)
  - Representation \( \Phi (x, y) \)
  - Algorithms to compute
    - \( \hat{y} = \arg\max_{y' \in Y} [w \cdot \Phi(x, y')] \)
    - \( \hat{y} = \arg\max_{y' \in Y} [\Delta (y', y) + w \cdot \Phi(x, y)] \)

  \( \rightarrow \) Generic structure covers OMM, MPD, Finite-State Transducers, MRF, etc.

Conditional Random Fields (CRF)

- Model:
  - \( P(y|x, w) = \frac{\exp(w \cdot \Phi(x, y))}{\sum_y \exp[w \cdot \Phi(x, y')]} \)
  - \( P(w) = \mathcal{N}(w|0, \lambda I) \)
- Conditional MAP training:
  \( \hat{w} = \arg\min_w [w \cdot w - \lambda \sum_i \log(P(y_i|x_i, w))] \)
- Prediction for zero/one loss:
  \( \hat{y} = \arg\max_y [w \cdot \Phi(x, y)] \)

Encoder/Decoder Networks

- Encoder: Build fixed-size representation of input sequence \( x \).
- Decoder: Generate output sequence \( y \) from encoder output.

Structured Prediction

- Discriminative ERM
  - Structural SVMs
  - Encoder/Decoder Nets
- Discriminative MAP
  - Conditional Random Fields
- Generative
  - Hidden Markov Model
- Other Methods
  - Maximum Margin Markov Networks
  - Markov Random Fields
  - Bayesian Networks
  - Statistical Relational Learning